

Claims

What is claimed is:

1 1. A Meta Model for building a knowledge-oriented software application using an object-
2 oriented programming language, the Meta Model comprising:

3 a Meta Knowledge Model comprising a specification of how the knowledge-oriented
4 software application represents knowledge;

5 a Meta Logic Model comprising a specification of how the knowledge-oriented
6 software application derives new knowledge;

7 a Knowledge Definition Model comprising a specification of how the knowledge-
8 oriented software application is converted from a human-readable format to
9 executable code;

10 a Catalog Model comprising a specification of how the knowledge-oriented software
11 application organizes the knowledge within an electronic device; and

12 a Run-time Model comprising a specification of how the knowledge-oriented
13 software application enables concurrent clients to manipulate the knowledge stored
14 in the electronic device.

1 2. The Meta Model of claim 1, wherein the Meta Knowledge Model further comprises a
2 Thing, a Key, a Fact, a Relation, and a Knowledge Definition.

1 3. The Meta Model of claim 2, wherein the Thing comprises an object-oriented
2 representation of a named object.

1 4. The Meta Model of claim 2, wherein the Key comprises an object-oriented
2 representation of a name of the Thing.

1 5. The Meta model of claim 2, wherein the Fact comprises an object-oriented
2 representation of facts regarding a set of things.

1 6. The Meta Model of claim 2, wherein the Relation further comprises an object-oriented
2 representation of a set of facts.

1 7. The Meta Model of claim 2, wherein the Knowledge Definition Model further
2 comprises an object oriented representation of a function for converting elements of the
3 Meta Model to an implementation.

1 8. The Meta Model of claim 1, wherein the Meta Logic Model further comprises an
2 Event, an Event Handler, and a Relation Function.

1 9. The Meta Logic Model of claim 8, wherein the Event comprises an object-oriented
2 representation of functions selected from a group consisting of: adding, editing,
3 modifying, and dropping a Relation.

1 10. The Meta Model of claim 8, wherein the Event Handler comprises an object oriented
2 representation of responses to the Event.

1 11. The Meta Model of claim 8, wherein the Relation Function comprises an object-
2 oriented representation of actions.

1 12. The Meta Model of claim 8, wherein the Relation Function comprises an object-
2 oriented representation of queries.

1 13. The Meta Model of claim 8, wherein the Relation Function comprises an object-
2 oriented representation of functions.

1 14. The Meta Model of claim 8, wherein the Relation Function receives a set of Relations
2 as input and returns a Relation as output.

1 15. A signal-bearing medium bearing a Meta Model comprising:
2 a Meta Logic Model comprising an Event, an Event handler, and a Relation
3 Function;

a Meta Knowledge Model comprising a Thing, a Key, a Fact, a Relation, and a Knowledge Definition;

a Knowledge Definition Model comprising a model for binding the Thing and its subtypes to a computer program;

a Catalog Model comprising a set of classes that are subclasses of the Relation, and wherein the Catalog Model indexes a set of knowledge definitions that are instances of subclasses of the Knowledge Definition Model; and

a Run-time Model comprising a set of classes that are subclasses of the Thing, wherein the Run-time Model dynamically binds the knowledge definitions to a physical implementation.

16. The signal-bearing medium of claim 15, wherein the Knowledge Definition Model further comprises a model for binding the Key and its subtypes, the Relation and its subtypes, the Fact and its subtypes, the Event and its subtypes, the Event Handler and its subtypes, and the Relation Function and its subtypes to the computer program.

17. The signal-bearing medium of claim 15, wherein the Knowledge Definition Model further comprises a Knowledge Definition and its subtypes.

18. The signal-bearing medium of claim 16, where the Knowledge Definition subtypes comprise a Thing Definition, a Key Definition, a Fact Definition, a Relation Definition, a Reference Definition, and a Relation Function Definition.

19. The signal-bearing medium of claim 18, wherein the Thing Definition comprises a logical name for the Thing and a name of a programming language class that implements the Thing.

20. The signal-bearing medium of claim 18, wherein the Key Definition comprises a logical name for the Key and a name of a programming language class that implements the Key.

- 1 21. The signal-bearing medium of claim 18, wherein the Fact Definition comprises a
2 logical name for the Fact and a name of a programming language class that implements
3 the Fact.
- 1 22. The signal-bearing medium of claim 18, where the Relation Definition comprises a
2 logical name for the Relation and a name of a programming language class that
3 implements the Relation.
- 1 23. The signal-bearing medium of claim 18, where the Relation Function Definition
2 comprises a logical name for the Relation Function and a name of a programming
3 language class that implements the Relation Function.
- 1 24. The signal-bearing medium of claim 18, wherein the Catalog Model comprises a
2 model for storing a declarative definition of an application and the application's run-time
3 data.
- 1 25. The signal-bearing medium of claim 24, wherein the Catalog Model further
2 comprising a set of classes that are subclasses of the Thing, Key, Relation, Fact, and
3 Knowledge Definition.
- 1 26. The signal-bearing medium of claim 24, wherein the Run-time Model comprises a
2 Session, a Kernel, and subtypes of an Application.
- 1 27. The signal-bearing medium of claim 24, wherein the Application comprises an object
2 oriented definition of how the application can be accessed by a computer program that
3 uses the services of the application.
- 1 28. The signal-bearing medium of claim 25, wherein the Session comprises an object
2 oriented definition of how the application can be accessed by concurrent users.
- 1 29. The signal-bearing medium of claim 26, wherein the Kernel comprises an object-
2 oriented definition of how multiple applications can be managed and accessed by
3 multiple users.

1 30. An electronic device comprising:
2 a processor; and
3 a storage device comprising:
4 a Meta Knowledge Model comprising a specification of how a knowledge-oriented
5 software application represents knowledge,
6 a Meta Logic Model comprising a specification of how the knowledge-oriented
7 software application derives new knowledge,
8 a Meta Knowledge Model comprising a Thing, a Key, a Fact, a Relation, and a
9 Knowledge Definition, wherein the Thing comprises an abstract base Thing class
10 with a method for retrieving a key of a Thing instance, wherein the key comprises a
11 unique identifier,
12 a Knowledge Definition Model comprising a specification of how the knowledge-
13 oriented software application is converted from a human-readable format to
14 executable code,
15 a Catalog Model comprising a specification of how the knowledge-oriented software
16 application organizes the knowledge within an electronic device, and
17 a Run-time Model comprising instructions that execute on the processor.

1 31. The electronic device of claim 30, wherein the Thing comprises a static Thing
2 Definition object that defines its physical implementation.

1 32. The electronic device of claim 31, wherein the Thing is extendible to create
2 application-specific Thing classes and wherein the Thing class is a base class for all
3 domain-specific object classes.

1 33. The electronic device of claim 30, wherein the Key comprises an abstract base type
2 and a StringKey, a DateKey and a IntegerKey are sub types of the Key.

- 1 34. The electronic device of claim 33, wherein the StringKey creates the Key from a
2 string and wherein the StringKey comprises a method to get a string value of a string key.
- 1 35. The electronic device of claim 33, wherein the IntegerKey creates the Key from an
2 integer and wherein the IntegerKey comprises a method to get an integer value of an
3 integer key.
- 1 36. The electronic device of claim 33, wherein the DateKey creates the Key from a date
2 and wherein the DateKey comprises a method to get a date value of a date key, and
3 wherein the Key is extendible to create other types of keys.
- 1 37. The electronic device of claim 30, where the Fact is a sub type of the Thing.
- 1 38. The electronic device of claim 30, wherein the Fact is as an abstract base type.
- 1 39. The electronic device of claim 30, wherein the Fact is a set of the Things, and
2 wherein the Things can be gotten and updated by their ordinal numbers.
- 1 40. The electronic device of claim 30, wherein the Fact is a set of the Things, and
2 wherein the Things can be gotten and updated by their names.
- 1 41. The electronic device of claim 30, wherein an implementation of the Fact has a fixed
2 cardinality.
- 1 42. The electronic device of claim 30, wherein the Fact has a Type and methods to get
2 and set the Fact Type, wherein the Fact Type can be Simple, OneToOne, and
3 ManyToOne.
- 1 43. The electronic device of claim 39, wherein the Things of the Fact can be gotten and
2 set as an array of the Things.
- 1 44. A signal-bearing medium comprising a Meta Model for building a knowledge-
2 oriented software application using an object-oriented programming language, wherein
3 the Meta Model comprises:

4 a Meta Knowledge Model including:

5 a Thing comprising a specification of how the knowledge-oriented software
6 application is represented as objects, wherein the Thing is the super class of the
7 objects;

8 a Key comprising a specification of how knowledge-oriented software application
9 things are named, where the thing names are instances of classes that are subclasses
10 of the Key and the things are instances of subclasses of the Thing;

11 a Fact comprising a specification of how knowledge oriented software application
12 represent facts, where the facts are instances of classes that are subclasses of the Fact
13 and the Fact is a subclass of the Thing,

14 a Relation comprising a specification of how the knowledge-oriented software
15 application represents sets of similar facts, where all relations are instances of classes
16 that are subclasses of the Relation, and the Relation is a subclass of Thing, and

17 a Knowledge Definition comprising a specification of how the knowledge-oriented
18 software application represents the logical design of the application and how it
19 relates the logical design to a physical implementation, wherein the Knowledge
20 Definition is a subclass of the Thing.

1 45. The signal-bearing medium of claim 44, wherein the Relation is a super type of the
2 Thing.

1 46. The signal-bearing medium of claim 44, wherein the Relation is as an abstract base
2 type.

1 47. The signal-bearing medium of claim 44, wherein the Relation is a set of named Facts.

1 48. The signal-bearing medium of claim 44, wherein the Relation has a Relation Scope.

1 49. The signal-bearing medium of claim 44, wherein the Relation has methods to get and
2 set the Relation Scope.

1 50. The signal-bearing medium of claim 48, wherein the Relation Scope comprises an
2 Application and a Session.

1 51. The signal-bearing medium of claim 44, where each instance of the Relation
2 comprises a name, and wherein the Relation has methods to get and set the name.

1 52. The signal-bearing medium of claim 44, wherein the Relation comprises methods to
2 retrieve, insert, delete, and update facts of the Relation.

1 53. The signal-bearing medium of claim 44, wherein Relation has methods to find the
2 union, intersections, and the difference of two relations.

1 54. The signal-bearing medium of claim 44, wherein the Relation comprises a method to
2 determine whether a fact is contained within a relation.

1 55. The signal-bearing medium of claim 44, wherein the Relation comprises a method to
2 get the number of facts in a relation.

1 56. The signal-bearing medium of claim 44, wherein the Relation comprises a method to
2 return all the facts as an array.

1 57. The signal-bearing medium of claim 44, wherein the Knowledge Definition is a sub
2 type of the Thing.

1 58. The signal-bearing medium of claim 44, wherein the Knowledge Definition is an
2 abstract base type.

1 59. The signal-bearing medium of claim 44, wherein the Knowledge Definition has a
2 name, and wherein the Knowledge Definition comprises a method to get and set the
3 name.

1 60. The signal-bearing medium of claim 44, wherein the Knowledge Definition
2 comprises a logical and physical representation of a Thing.

1 61. A signal-bearing medium comprising a Meta Model for computer programs
2 implemented in an object-oriented programming language, the Meta Model comprising:

3 a Meta Knowledge Model comprising a Thing, a Key, a Fact, a Relation, and a
4 Knowledge Definition; and

5 a Meta Logic Model comprising an Event, an Event Handler, and a Relation
6 Function, wherein the Event is a sub type of the Thing.

1 62. The signal-bearing medium of claim 61, wherein the Event has a name, and wherein
2 the Event has methods to get and set the name.

1 63. The signal-bearing medium of claim 61, wherein the Event has a Level and wherein
2 the Event has methods to get and set the Event Level, and wherein the Event Level can be
3 selected from a group consisting of: the Relation and the Fact.

1 64. The signal-bearing medium of claim 61, wherein the Event has a Priority, and
2 wherein the Event has methods to get and set the Event Priority.

1 65. The signal-bearing medium of claim 61, wherein the Event has a Scope, and wherein
2 the Event has methods to get and set the Event Scope, and wherein the Event Scope can
3 be selected from a group consisting of: an Application or a Session.

1 66. The signal-bearing medium of claim 61, wherein the Event has a Type, wherein the
2 Event has methods to get and set the Event Type, and wherein the Event Type can be
3 selected from a group consisting of: On Insert, On Delete, On Update, On Add, and On
4 Drop.

1 67. The signal-bearing medium of claim 61, wherein the Event has methods to get and set
2 the Event Handler.

1 68. The signal-bearing medium of claim 61, wherein the Event Handler comprises a sub
2 type of the Thing.

1 69. The signal-bearing medium of claim 61, wherein the Event Handler comprises:

2 a name; and

3 methods to get and set the name.

1 70. The signal-bearing medium of claim 61, wherein the Event Handler comprises:

2 a relation function; and

3 methods to get and set a name of the relation function.

1 71. The signal-bearing medium of claim 61, wherein the Event Handler comprises:

2 a target relation; and

3 methods to get and set a name of the target relation.

1 72. The signal-bearing medium of claim 61, wherein the Event Handler comprises:

2 an action type; and

3 methods to get and set the action type, wherein the action type is selected from a
4 group consisting of: Clear, Add, Drop, Replace, Insert, Delete, and Update.

1 73. The signal-bearing medium of claim 61, wherein the Relation Function is a super type
2 of the Thing.

1 74. The signal-bearing medium of claim 61, wherein the Relation Function comprises:

2 a name; and

3 methods to get and set the name.

- 1 75. The signal-bearing medium of claim 61, wherein the Relation Function has an
2 argument and the argument is an array of the Relations.
- 1 76. The signal-bearing medium of claim 61, wherein the Relation Function has a return
2 value and the return value is a Relation.
- 1 77. The signal-bearing medium of claim 61, wherein the Relation Function defines a
2 method to execute.
- 1 78. The signal-bearing medium of claim 61, wherein the Relation Function and the
2 Relation form a closed algebra, wherein the Relation Function returns only a relation, and
3 wherein the Relation Function can be used as an argument anywhere a Relation can be
4 used as an argument.
- 1 79. The signal-bearing medium of claim 61, wherein the Relation Function is recursive,
2 wherein an output of the Relation Function is an input of the Relation Function, and
3 wherein the Relation Function terminates when an output Relation is empty.
- 1 80. A signal-bearing medium comprising:
2 a Meta Model for computer programs implemented in an object-oriented
3 programming language, the Meta Model comprising:
4 a Meta Knowledge Model comprising a Thing, a Key, a Fact, a Relation,
5 and a Knowledge Definition, and
6 a Meta Logic Model comprising an Event, an Event Handler, and a
7 Relation Function; and
8 a Kernel, wherein the Kernel is a super type of the Thing.
- 1 81. The signal-bearing medium of claim 80, wherein the Kernel comprises a system
2 Application that manages external applications.

- 1 82. The signal-bearing medium of claim 80, wherein the Kernel comprises a name and
2 methods to get and set the name.
- 1 83. The signal-bearing medium of claim 80, wherein the Kernel comprises a root object
2 of all objects.
- 1 84. The signal-bearing medium of claim 80, wherein the Kernel has a number of users
2 and methods to add and drop users identified by their names and passwords.
- 1 85. The signal-bearing medium of claim 80, wherein the Kernel grants access to
2 Applications and wherein the Kernel comprises methods to grant privileges to users.
- 1 86. The signal-bearing medium of claim 80, wherein the Kernel has multiple concurrent
2 users, wherein each of the users has multiple sessions, and wherein the Kernel comprises
3 methods to create, get, and drop named sessions.
- 1 87. The signal-bearing medium of claim 80, wherein the Kernel comprises methods to
2 load and unload an Application.
- 1 88. The signal-bearing medium of claim 80, wherein the Kernel loads an application from
2 an Application Definition File.
- 1 89. The signal-bearing medium of claim 80, wherein the signal-bearing medium further
2 comprises an Application.
- 1 90. The signal-bearing medium of claim 89, wherein the Application is a sub type of the
2 Thing.
- 1 91. The signal-bearing medium of claim 89, wherein the Application has a version and
2 the Application has methods to get and set the version.
- 1 92. The signal-bearing medium of claim 89, wherein the Application has a name and the
2 Application has methods to get and set the name.

1 93. The signal-bearing medium of claim 89, wherein the Application has a set of Relation
2 Definition instances and the Application has methods to add, get, and drop the Relation
3 Definition instances.

1 94. The signal-bearing medium of claim 89, wherein the Application has a set of Relation
2 instances and the Application has methods to add, get, and drop the Relation instances.

1 95. The signal-bearing medium of claim 89, wherein the Application has a set of Relation
2 Function Definition instances and the Application has methods to add, get, and drop the
3 Relation Function Definition instances.

1 96. The signal-bearing medium of claim 89, wherein the Application has a set of Relation
2 Function instances and the Application has methods to add, get, and drop the Relation
3 Function instances.

1 97. The signal-bearing medium of claim 89, wherein the Application has a set of Event
2 instances and the Application has methods to add, get, and drop the Event instances.

1 98. The signal-bearing medium of claim 89, wherein the Application has a set of Event
2 Handler instances and the Application has methods to add, get, and drop the Event
3 Handler instances.

1 99. The signal-bearing medium of claim 89, wherein the Application has a set of Thing
2 Definition instances and the Application has methods to add, get, and drop the Thing
3 Definition instances.

1 100. The signal-bearing medium of claim 80, wherein the Kernel enforces type safety.

1 101. The signal-bearing medium of claim 100, wherein when a relation, relation function,
2 thing, key, and fact are created and referenced, the Kernel enforces the type specified in a
3 corresponding knowledge definition.

1 102. The signal-bearing medium of claim 80, wherein the Kernel enforces referential
2 integrity.

1 103. The signal-bearing medium of claim 102, where when a relation, relation function,
2 thing, key, and fact are referenced by the key, the Kernel enforces that an object
3 referenced by the key exists in the relation specified by the key's reference definition.

1 104. The signal-bearing medium of claim 89, wherein the Application manages a
2 transaction, wherein a transaction comprises a change in a state of a relation.

1 105. The signal-bearing medium of claim 104, wherein the Application has a method to
2 start the transaction.

1 106. The signal-bearing medium of claim 104, wherein the Application has a method to
2 commit the transaction.

1 107. The signal-bearing medium of claim 104, wherein the Application has a method to
2 rollback the transaction.

1 108. The signal-bearing medium of claim 104, wherein the Application has a method to
2 execute the Relation Function and return a result as a relation.

1 109. The signal-bearing medium of claim 104, wherein the Application rollbacks an
2 external transaction when the external transaction fails.

1 110. The signal-bearing medium of claim 109, wherein the Application rollbacks an
2 external transaction when a transaction of an event generated by the external transaction
3 fails.

1 111. The signal-bearing medium of claim 109, wherein the Application commits the
2 external transaction and the transactions generated by the events as a result of the external
3 transaction.

1 112. The signal-bearing medium of claim 89, wherein the Application is persistently and
2 declaratively stored in an Application Definition File.

- 1 113. The signal-bearing medium of claim 112, wherein a Kernel binds the Application
2 Definition to a physical implementation at run time.
- 1 114. The signal-bearing medium of claim 80, wherein a Session is a sub type of the
2 Thing.
- 1 115. The signal-bearing medium of claim 114, wherein the Session has a name and the
2 Session has methods to get and set the name.
- 1 116. The signal-bearing medium of claim 114, wherein the Session has methods to access
2 any Kernel application.
- 1 117. The signal-bearing medium of claim 114, wherein the Session has a time out and
2 wherein the Kernel removes timed-out sessions automatically.
- 1 118. The signal-bearing medium of claim 114, wherein the Session has a state.